This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1- 14. (Canceled)

15.   (Currently amended) A method for streamlining operations in a coprocessor by ~~indicating whether~~ ensuring all required memory resources for a task are available, comprising:

prior to beginning the processing of the task,

preparing ~~a~~ the task for processing in a coprocessor by paging memory resources, within a memory space accessible by the coprocessor, associated with the task into coprocessor-readable memory;

sampling the memory resources to determine if all required memory resources are in a proper location in the coprocessor-readable memory; and

recording whether all required memory resources are in a proper location in the coprocessor-readable memory, wherein said recording generates an indicator memory resource that is associated with the task; ~~and~~

processing said indicator memory resource substantially at the beginning of processing the task, wherein ~~if~~ when said indicator memory resource indicates that all required memory resources are not in the proper location in the coprocessor-readable memory, a first page fault is generated, and the coprocessor stops ~~the initiation of~~ beginning the processing of the task, wherein a list of tasks is maintained that comprises tasks that the coprocessor stopped processing; and

generating a second page fault when a context switch occurs to a task that references an invalid ring buffer or an invalid DMA buffer.

16.   (Previously Presented) A method according to claim 15 wherein the coprocessor is a Graphics Processing Unit (GPU).

17.   (Original)   A method according to claim 15 wherein the task is represented by a DMA buffer.

18.   (Canceled)

19.   (Currently Amended) A method according to claim 15, further comprising ~~maintaining a~~ <u>utilizing said</u> list ~~of tasks that the coprocessor stopped processing, so that all~~ <u>to</u> <u>bring</u> required memory resources ~~can be brought~~ to a proper location in coprocessor-readable memory at a later time.

20.   (Original)   A method according to claim 19 wherein the later time is determined based on a priority of tasks on the list of tasks.

21.   (Original)   A method according to claim 20, further comprising a periodic priority boost that increases the priority of one or more tasks on the list of tasks to ensure that all tasks eventually can be processed.

22-37. (Canceled)

38.   (Currently Amended) A method for streamlining operations in a coprocessor by ~~indicating whether~~ <u>ensuring</u> all required memory resources for a task are available, comprising:

prior to beginning the processing of the task,

preparing the task for processing in a Graphics Processing Unit (GPU) by paging memory resources within a memory space accessible by the GPU associated with the task into GPU readable memory;

sampling the memory resources to determine if all required memory resources are in a proper location in the GPU readable memory; <u>and</u>

recording whether all required memory resources are in a proper location in the GPU readable memory, wherein said recording generates an indicator memory resource that is associated with the task; ~~and~~

processing said indicator memory resource prior to beginning processing the task, wherein ~~if~~ <u>when</u> said indicator <u>memory</u> resource indicates that all required memory resources

are not in the proper location in the GPU readable memory, a ~~first~~ page fault is generated, the task is not processed and <u>wherein</u> a list ~~comprising~~ <u>of tasks is maintained that comprises</u> tasks that the coprocessor stopped processing ~~is maintained~~<u>; and</u>

<u>generating a second page fault when a context switch occurs to a task that references an invalid ring buffer or an invalid DMA buffer.</u>

39.   (Previously Presented) A method according to claim 38 wherein the task is represented by a DMA buffer.

40.   (Canceled)

41.   (Previously Presented) A method according to claim 38 wherein the later time is determined based on a priority of tasks on the list of tasks.

42.   (Previously Presented) A method according to claim 41, further comprising a periodic priority boost that increases the priority of one or more tasks on the list of tasks to ensure that all tasks eventually can be processed.

43.   (Canceled)